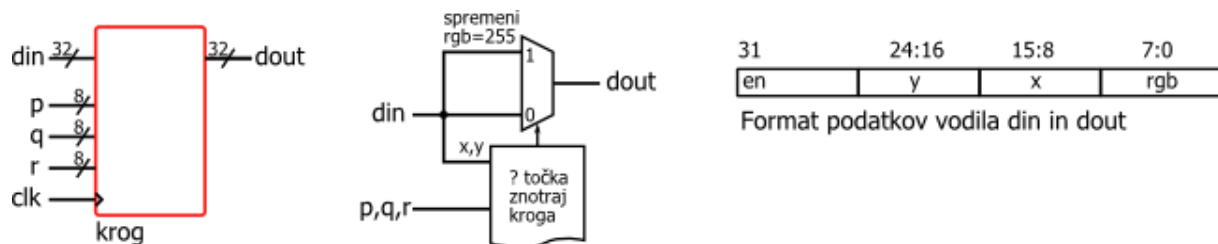


1. vaja: Krog

Naredili bomo komponento vezja, ki izračuna ali se točka na koordinati (x,y) nahaja znotraj kroga s parametri (p,q,r) in nastavi barvo točk. Koordinate točk in barva se prenašajo prek vodil **din** in **dout**, parametre kroga pa določajo zunanji signali. Komponento bomo povezali z vmesnikom AXI in naredili IP (komponento intelektualne lastnine).

1.1 Opis vezja: krog.vhd



Naloga vezja je obdelava slikovnih točk, ki se prenašajo v vezje po 32-bitnem vodilu **din** in gredo iz vezja po vodilu **dout**. Iz podatkov na vodilu **din** izluščimo koordinati vhodne točke x in y , ki sta v območju $0..255$, **rgb** pa predstavlja barvo točke. Signali **p,q** in **r** so 8-bitni parametri (vrednosti $0..255$), ki določajo ali se vhodna slikovna točka nahaja znotraj kroga:

$$(x-p)^2 + (y-q)^2 < r^2$$

Če je neenačba izpolnjena, naj bo vrednost barve na izhodu 255 (bela), sicer pa naj bo izhodna točka enaka vhodni. Pri opisu v jeziku VHDL moramo paziti na ustrezno pretvorbo podatkovih tipov in velikosti vektorjev. Npr. za izračun razlike $(x-p)$ najprej pretvorimo obe vhodni vrednosti v 9-bitno predznačeno število (**signed**), potem pa izračunamo razliko. Primer pretvorbe:

```
x <= '0' & signed(din(15 downto 8));
p1 <= '0' & signed(p);
```

Koraki izdelave projekta v programu Vivado

- V programu Vivado naredi nov projekt, določi ime (npr. krog) in lokacijo na disku (D:\div\moja mapa),
- nato izberi **RTL Project**, v naslednjem oknu pa klikni + in **Create File** ter določi ime datoteke (krog) in vrsto: VHDL.
- Možnost dodajanja obstoječih komponent (**Add Existing IP**) v projekt in datotek navodili za prevajanje (**Add Constraints**) pustimo prazno.
- V zadnjem oknu izberemo FPGA ali razvojno ploščo: klikni **Select: Boards** in izberi razvojno ploščo *ZedBoard Zynq Evaluation and Development Kit*.
- Pred zaključkom se pojavi še okno v katerem določi vhodne in izhodne signale nove VHDL datoteke.

Ob vsakem shranjevanju datoteke z opisom vezja (ctrl+s) program avtomatsko preveri osnovna sintaktična pravila. V primeru napake dobimo sporočila v zavihku **Messages** in datoteka se v hierarhiji projekta premakne v mapo Syntax Error Files. Dokler so napake in je datoteka v tej mapi, ne moremo izvajati naslednjih korakov.

Shematsko strukturo vezja (RTL) lahko prikažemo takoj, ko shranimo VHDL datoteko, tako da izberemo iz menija **Flow** opcijo **Open Elaborated Design**

Po opisu vezja naredimo:

- Sintezo vezja (v oknu na levi kliknemo **Run Synthesis**)
 - Po sintezi izberemo **View Reports** in si ogledamo poročila: **Vivado Synthesis Report** in **Utilization Report**. V prvem poročilu vidimo število in vrsto RTL gradnikov in število celic v vezju, v drugem poročilu pa zasedenost FPGA matrice (npr. število uporabljenih vpoglednih tabel LUT).
- Simulacijo vezja (**Run Simulation, Run Behavioral Simulation**)
 - V simulatorju nastavimo vrednosti vhodnim signalom (**Force Constant** ali **Clock**) in opazujemo izhode.

Vsi koraki prevajanja puščajo kratka opozorila in sporočila o napakah v zavihku **Messages**, podrobnejše informacije o napaki pa vidimo v zavihku **Tcl Console**.

1.2 Komponenta AXI



- Izdelavo nove komponente IP začnemo z odprtim projektom, npr. Krog, ki mora imeti nastavljen privzeti jezik VHDL (preveri **Project Settings**)
- Izberi **Tools > Create and Package IP**, nato **Next** in izberi **Create a new AXI4 peripheral**
- Določi ime: AXI_krog in mapo v kateri bodo datoteke (npr. AXI_krog_ip)
- V naslednjem oknu le še potrdi nastavitve: Interface: Lite, Slave, Data Width: 32, Number of registers: 4
- V zadnjem oknu pa izberi opcijo: **Edit IP**, ki odpre nov projekt

Sedaj bomo na kratko zapustili program Vivado in si pripravili datoteke v sistemu. Datoteke izdelane komponente se nahajajo v mapi: AXI_krog_ip\AXI_krog_1.0, VHDL datoteke so v podmapi **hdl**. V to mapo najprej kopiraj datoteko krog.vhd, nato pa jo vključi v projekt programa Vivado (**Add Sources**). **Zaradi kasnejše uporabe nove komponente IP je zelo pomembno, da so vse izvorne datoteke zbrane na enem mestu, ne pa vključene kot povezave na druge mape !**

Odpri AXI_krog_v1_0.vhd, dodaj 32-bitna priključka **din** in **dout** ter deklaracijo komponente **krog**. Nato odpri še AXI_krog_v1_0_S00_AXI.vhd in dodaj 8-bitne izhodne priključke **p**, **q** in **r**, ki jih v arhitekturnem delu poveži na signale **slv_reg2**, **slv_reg1** in **slv_reg0**. Popravi deklaracijo te komponente v AXI_krog_v1_0.vhd in naredi povezave s komponento **krog**.